CLAIMS

1. A method for processing a workflow, said workflow including a plurality of activities and workflow transition information, said workflow being processed by a plurality of activity servers, each of said plurality of activity servers performing at least one of said plurality of activities, the method comprising the steps of:

a) retrieving from a workflow queue a workflow packet requiring one of said plurality of activities to be executed, said workflow packet being retrieved by one of said plurality of activity servers performing said one of said plurality of activities;

b) executing said activity, said activity being executed by said one of said plurality of activity servers;

c) determining a next activity to be performed based on said workflow transition information;

d) executing said next activity if said one of said plurality of activity servers performs said next activity

e) forwarding to said workflow queue said next activity if said one of said plurality of activity servers does not perform said activity

repeating steps a – e until all of said plurality of activities in said workflow are executed.

2. The method of claim 1, wherein said workflow packet includes a process state.

3. The method of claim 1, further comprising a database for storing said transition information, said method further comprising the step of:

retrieving said transition information from said database, said transition information being retrieved by all of said plurality of activity servers.

4. The method of claim 1, wherein the step of forwarding to said workflow queue includes the steps of:

persisting said workflow packet requiring said next activity and said next process state; and

forwarding said workflow packet to said workflow queue for one of the plurality of activity servers providing said next activity.

5. The method of claim 1, wherein at least one of said plurality of activity servers performs more than one of said plurality of activities.

6. The method of claim 1, wherein at least one of said plurality of activities is an automatic activity.

7. The method of claim 1, wherein at least one of said plurality of activities is a manual activity for receiving an input from a user.

8. The method of claim 7, wherein said manual activity manages a user interface with said user.

9. The method of claim 1, wherein more than one of said plurality of activities is a manual activity and wherein said more than one of said plurality of activities are aggregated in one of said plurality of activity servers.

10. The method of claim 9, wherein said one of said plurality of activity servers interfaces with a desktop server for providing a user interface to a user.

11. The method of claim 1, wherein at least one of said plurality of activities is performed by more than one of said plurality of activity servers.

12. The method of claim 1, further comprising the step of:

receiving an event notification requesting that said workflow be processed; and

initiating said workflow.

13. The method of claim 1, wherein said transition information includes a routing transition.

14. The method of claim 13, further comprising the step of:

performing more than one of said plurality of activities and more than one routing transition in a single transaction in one of said plurality of activity servers.

15. The method of claim 1, wherein said transition information includes a route number, a node number, a routing transition and a next node number.

16. A system for processing a workflow, said workflow including a plurality of activities and workflow transition information, comprising:

a database for storing said workflow transition information;

a workflow queue for storing a plurality of workflow packets including at least one of said plurality of activities; and

a plurality of activity servers, each of said plurality of activity servers performing at least one of said plurality of activities; each of said plurality of activity servers including a workflow engine for receiving said workflow transition information from said database;

a plurality of resource managers, each of said plurality of activity servers including one of said plurality of resource managers;

wherein each of said plurality of resource managers retrieves from said workflow queue at least one of said plurality of workflow packets that requires one of said plurality of activities for execution by said corresponding one of said plurality of activity servers, wherein each of said plurality of resource managers determines a next activity to be performed in said workflow based on said transition information and wherein each of said plurality of resource managers forwards

to said workflow queue said next activity if said corresponding one of said plurality of activity servers does not perform said next activity.

17. The system of claim 16, wherein said workflow packet includes a process state.

18. The system of claim 16, wherein each of said plurality of resource managers forwards the workflow packet including said next activity and said next process state and forwards said workflow packet to said workflow queue.

19. The system of claim 16, wherein at least one of said plurality of activity servers performs more than one of said plurality of activities.

20. The system of claim 16, wherein at least one of said plurality of activities is an automatic activity.

21. The system of claim 16, wherein at least one of said plurality of activities is a manual activity for receiving an input from a user.

22. The system of claim 21, wherein said manual activity manages a user interface with said user.

23. The system of claim 16, wherein more than one of said plurality of activities is a manual activity and wherein said more than one of said plurality of activities are aggregated in one of said plurality of activity servers.

24. The system of claim 23, wherein said one of said plurality of activity servers interfaces with a desktop server for providing a user interface to a user.

25. The method of claim 16, wherein at least one of said plurality of activities is performed by more than one of said plurality of activity servers.

26. The system of claim 16, further comprising an event receiver, said event receiver receiving an event notification for initiating said workflow.

27. The system of claim 16, wherein said transition information includes a route number, a node number, a routing transition and a next node number.

28. Computer executable program code residing on a computer-readable medium, the program code comprising instructions for causing the computer to:

perform a method for processing a workflow, said workflow including a plurality of activities and workflow transition information, said workflow being processed by a plurality of activity servers, each of said plurality of activity servers performing at least one of said plurality of activities;

a) retrieve from a workflow queue a workflow packet including one of said plurality of activities to be executed, said workflow packet being retrieved by one of said plurality of activity servers performing said one of said plurality of activities;

b) execute said activity, said activity being executed by said one of said plurality of activity servers;

c) determine a next activity to be performed based on said workflow transition information;

d) execute said next activity if said one of said plurality of activity servers performs said next activity;

e) forward to said workflow queue said next activity if said one of said plurality of activity servers does not perform said activity; and

repeat steps a – e until all of said plurality of activities in said workflow are executed.

29. The computer executable program of claim 28, wherein said workflow packet includes a process state.

30. The computer executable program of claim 28, further comprising a database for storing said transition information, and wherein the program code additionally causes the computer to:

retrieve said transition information from said database, said transition information being retrieved by all of said plurality of activity servers.

31. The computer executable program of claim 28, wherein the program code additionally causes the computer to:

form a next workflow packet including said next activity and said next process state; and

forward said next workflow packet to said workflow queue.

32. The computer executable program of claim 28, wherein at least one of said plurality of activity servers performs more than one of said plurality of activities.

33. The computer executable program of claim 28, wherein at least one of said plurality of activities is an automatic activity.

34. The computer executable program of claim 28, wherein at least one of said plurality of activities is a manual activity for receiving an input from a user.

35. The computer executable program of claim 34, wherein said manual activity manages a user interface with said user.

36. The computer executable program of claim 28, wherein more than one of said plurality of activities is a manual activity and wherein said more than one of said plurality of activities are aggregated in one of said plurality of activity servers.

37. The computer executable program of claim 36, wherein said one of said plurality of activity servers interfaces with a desktop server for providing a user interface to a user.

38. The computer executable program of claim 28, wherein at least one of said plurality of activities is performed by more than one of said plurality of activity servers.

39. The computer executable program of claim 28, wherein the program code additionally causes the computer to:

receive an event notification requesting that said workflow be processed; and

initiate said workflow.

40. The computer executable program of claim 28, wherein said transition information includes a route number, a node number, a routing transition and a next node number.